



Siemens PLM Software

# HEEDS MDO 2019.1 Setting up a Windows-to- Windows Compute Resource

[www.redcedartech.com](http://www.redcedartech.com)

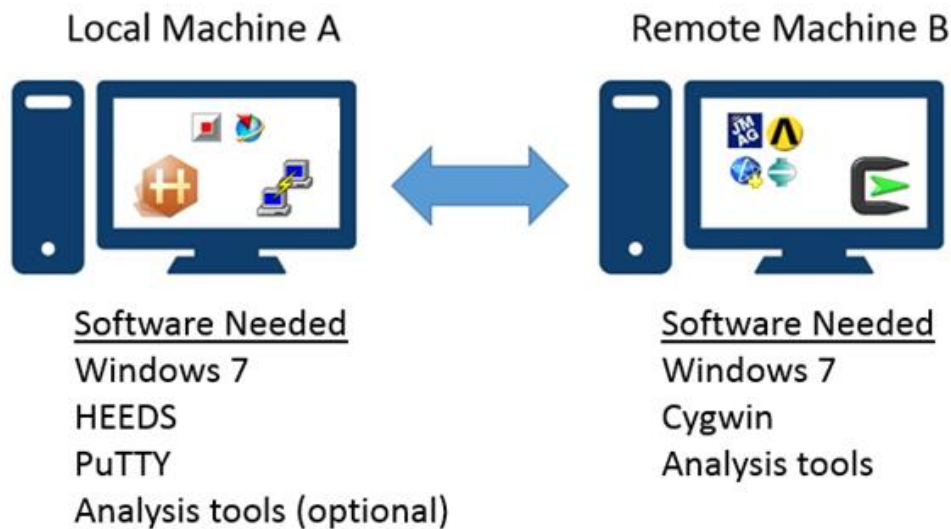
# Contents

---

<b>INTRODUCTION.....</b>	<b>1</b>
<b>ON REMOTE MACHINE B.....</b>	<b>2</b>
INSTALLING CYGWIN .....	2
STARTING THE SSH SERVER .....	4
ENABLING SSH THROUGH THE WINDOWS FIREWALL .....	6
<b>ON LOCAL MACHINE A .....</b>	<b>9</b>
INSTALLING PUTTY .....	9
<b>TESTING THE CONNECTION BETWEEN MACHINES.....</b>	<b>10</b>
<b>CREATING PASSWORD-LESS ENTRY TO REMOTE MACHINE B USING RSA KEYS .....</b>	<b>11</b>
CREATE PRIVATE AND PUBLIC KEYS ON LOCAL MACHINE A .....	11
COPY PUBLIC KEY TO REMOTE MACHINE B.....	14
ENABLE AND TEST RSA KEY ACCESS WITH PAGEANT ON LOCAL MACHINE A .....	15
<b>SETTING UP REMOTE EXECUTION IN HEEDS .....</b>	<b>17</b>
CREATING A REMOTE PROFILE .....	17
CREATING AND ASSIGNING A COMPUTE RESOURCE.....	18
<b>NOTES .....</b>	<b>19</b>
<b>TROUBLESHOOTING.....</b>	<b>20</b>
SSH DOESN'T CONNECT TO REMOTE MACHINE.....	20
PASSWORD-LESS CONNECTION IS UNSUCCESSFUL.....	20
SETTING UP A LOCAL SHARED DRIVE RESOURCE .....	21
<b>EXAMPLE STUDY .....</b>	<b>23</b>
HEEDS EXAMPLE STUDY EXECUTION .....	23

# Introduction

This document provides a step-by-step guide to establishing communication between two machines, one of which is running HEEDS MDO, each running a Windows operating system. Once this communication has been established, HEEDS can then operate on a local Windows workstation while submitting jobs to a remote Windows workstation, as illustrated in the figure below. Some of the steps defined in this document require administrative privileges on each Windows workstation. Before proceeding, please ensure that you have sufficient privileges to complete the required tasks.



A network protocol is required to allow remote login and other network services to operate securely over an unsecured network. The Secure SHell, or SSH, cryptographic network protocol is employed in a client-server architecture for the communications described in this document. The HEEDS team strongly recommends utilizing the free, open-source, network file transfer application called PuTTY as the SSH client on Local Machine A and the Unix-like environment for Windows, called Cygwin, as the SSH server on Remote Machine B to set up a Windows-to-Windows compute resource.

# On Remote Machine B



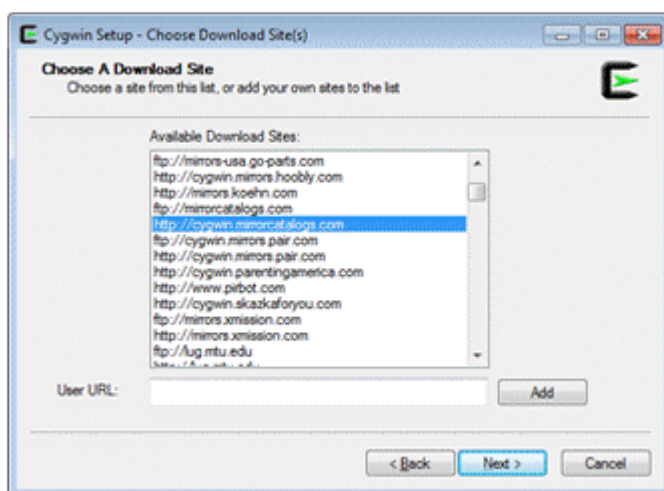
## Installing Cygwin

Cygwin serves as the SSH server for the remote machine. It facilitates the ability for HEEDS to send data to and from the remote machine. It is a free downloadable program that gives a Unix-like front-end to the Windows remote machine. All HEEDS commands passed to the remote machine therefore will be Unix syntax in nature. To download and configure Cygwin such that HEEDS can interact remotely with Machine B:

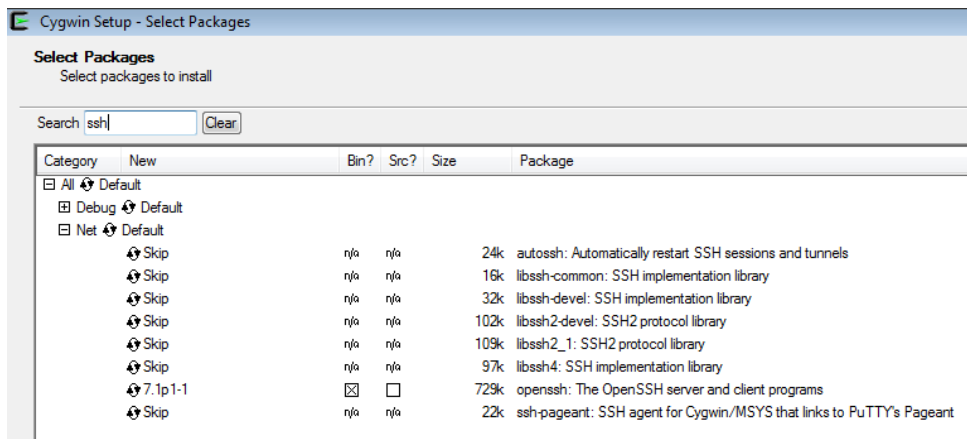
1. Navigate your web browser to <https://cygwin.com/install.html>. Click to download the appropriate version of Cygwin for your Windows workstation (32-bit or 64-bit operating system).



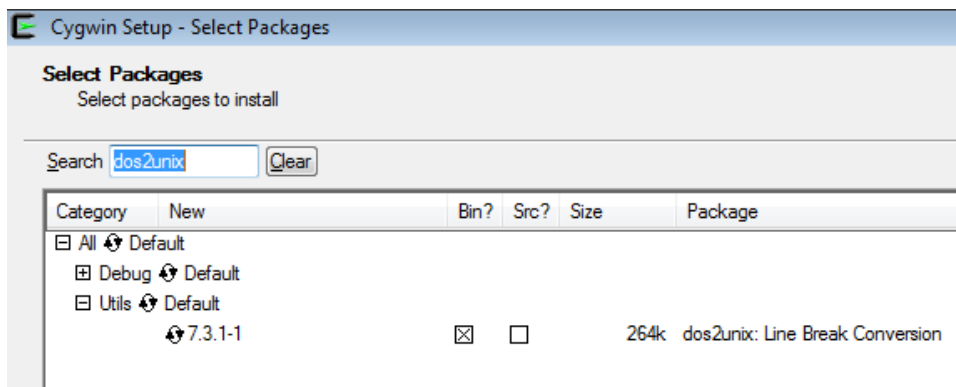
2. Run the installer and choose **Install from Internet**. Use the default options for each screen until asked to choose your **Download Site**.
3. Choose a download site from the presented list, preferring those that are closer to your location and that use the faster protocol (http versus ftp).



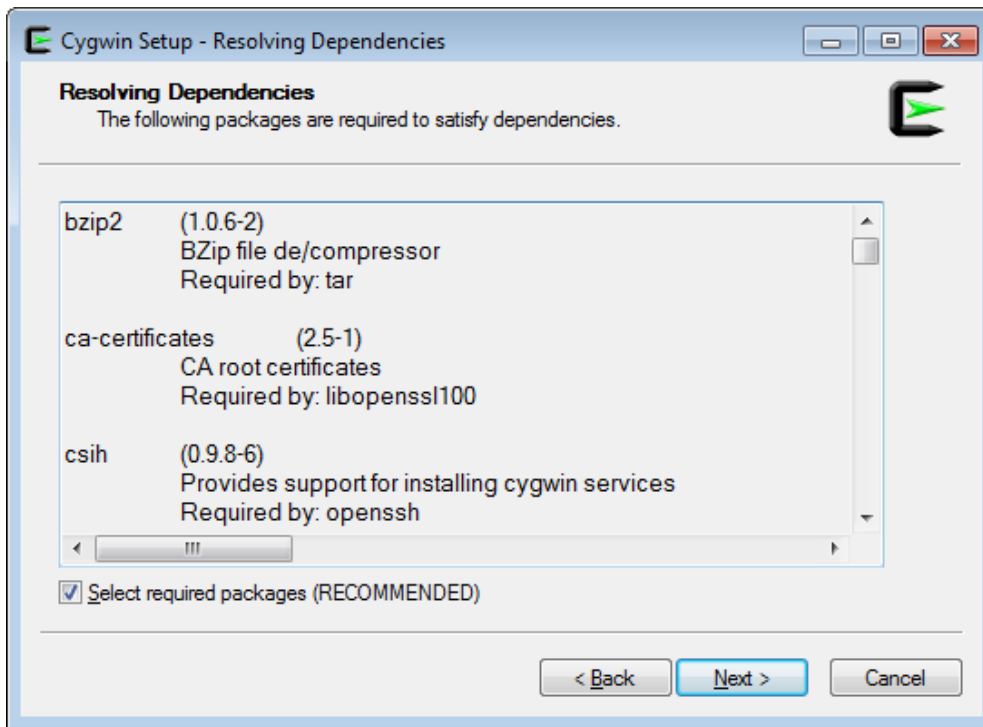
4. The next window will ask you to select packages of Cygwin to install. Only two packages are required: **openssh** and **dos2unix**. In the search bar at the top of the screen, type **ssh**, and then click on the **+** next to **Net**. Click once on the package titled **openssh** (click directly on **Skip** to select the package). Your screen should appear as shown next.



5. Follow the same process as step 4, this time searching for the package **dos2unix**. Expand the **+** next to **Utils**, and click once on the package titled **dos2unix**. Your screen should appear as shown next.



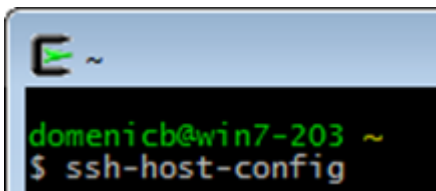
- Click **Next** after both packages have been selected for installation. Click **Next** at the bottom of the **Resolving Dependencies** window after ensuring that the **Select required packages** option is checked. Cygwin and all necessary packages for HEEDS remote execution should now be installed.



## Starting the SSH Server

Now that Cygwin is installed, we need to start an SSH Server on it. This is the actual mechanism that allows HEEDS to pass data to the machine. Without starting the server successfully HEEDS will not be able to pass data or execute analyses on the remote machine.

- Once the installation is complete, you will need to run Cygwin as an administrator. To do this, right click on the Cygwin icon and choose **Run as administrator**.
- When Cygwin is launched, you should see a command prompt window with a single line of text stating **username@machinename**. On the following line, type **ssh-host-config** and press **Enter**. This step sets up the configuration file for the SSH server.



- The resulting output in the Cygwin window is as seen below. You will be prompted to answer several questions during this output. Enter the responses highlighted in **yellow** below. If Cygwin was previously installed on the machine, the number of initial questions requiring a “yes” command may be different. This is normal.

```
$ ssh-host-config
```

```
*** Info: Generating missing SSH host keys
ssh-keygen: generating new host keys: RSA DSA ECDSA ED25519
*** Info: Creating default /etc/ssh_config file
```

```
*** Info: Creating default /etc/sshd_config file

*** Info: StrictModes is set to 'yes' by default.
*** Info: This is the recommended setting, but it requires that the POSIX
*** Info: permissions of the user's home directory, the user's .ssh
*** Info: directory, and the user's ssh key files are tight so that
*** Info: only the user has write permissions.
*** Info: On the other hand, StrictModes don't work well with default
*** Info: Windows permissions of a home directory mounted with the
*** Info: 'noacl' option, and they don't work at all if the home
*** Info: directory is on a FAT or FAT32 partition.
*** Query: Should StrictModes be used? (yes/no) yes

*** Info: Privilege separation is set to 'sandbox' by default since
*** Info: OpenSSH 6.1. This is unsupported by Cygwin and has to be set
*** Info: to 'yes' or 'no'.
*** Info: However, using privilege separation requires a non-privileged account
*** Info: called 'sshd'.
*** Info: For more info on privilege separation read
/usr/share/doc/openssh/README.privsep.
*** Query: Should privilege separation be used? (yes/no) yes
*** Info: Note that creating a new user requires that the current account have
*** Info: Administrator privileges. Should this script attempt to create a
*** Query: new local account 'sshd'? (yes/no) yes
*** Info: Updating /etc/sshd_config file

*** Query: Do you want to install sshd as a service?
*** Query: (Say "no" if it is already installed as a service) (yes/no) yes
*** Query: Enter the value of CYGWIN for the daemon: [] ntsec tty
*** Info: On Windows Server 2003, Windows Vista, and above, the
*** Info: SYSTEM account cannot setuid to other users -- a capability
*** Info: sshd requires. You need to have or to create a privileged
*** Info: account. This script will help you do so.

*** Info: It's not possible to use the LocalSystem account for services
*** Info: that can change the user id without an explicit password
*** Info: (such as passwordless logins [e.g. public key authentication]
*** Info: via sshd) when having to create the user token from scratch.
*** Info: For more information on this requirement, see
*** Info: https://cygwin.com/cygwin-ug-net/ntsec.html#ntsec-nopasswd1

*** Info: If you want to enable that functionality, it's required to create
*** Info: a new account with special privileges (unless such an account
*** Info: already exists). This account is then used to run these special
*** Info: servers.

*** Info: Note that creating a new user requires that the current account
*** Info: have Administrator privileges itself.

*** Info: No privileged account could be found.
```

```

*** Info: This script plans to use 'cyg_server'.
*** Info: 'cyg_server' will only be used by registered services.
*** Query: Do you want to use a different name? (yes/no) yes
*** Query: Enter the new user name: cyg_server
*** Query: Reenter: cyg_server

*** Query: Create new privileged user account 'RCT-127\cyg_server' (Cygwin
name: 'cyg_server')? (yes/no) yes
*** Info: Please enter a password for new user cyg_server. Please be sure
*** Info: that this password matches the password rules given on your system.
*** Info: Entering no password will exit the configuration.
*** Query: Please enter the password: password
*** Query: Reenter: password

*** Info: User 'cyg_server' has been created with password 'password'.
*** Info: If you change the password, please remember also to change the
*** Info: password for the installed services which use (or will soon use)
*** Info: the 'cyg_server' account.

*** Info: The sshd service has been installed under the 'cyg_server'
*** Info: account. To start the service now, call `net start sshd' or
*** Info: `cygrunsrv -S sshd'. Otherwise, it will start automatically
*** Info: after the next reboot.

*** Info: Host configuration finished. Have fun!

```

4. Type the command **net start sshd** and press **Enter**. This step starts the SSH server. The result should look as shown next.

```

$ net start sshd
The CYGWIN sshd service is starting.
The CYGWIN sshd service was started
successfully.

```

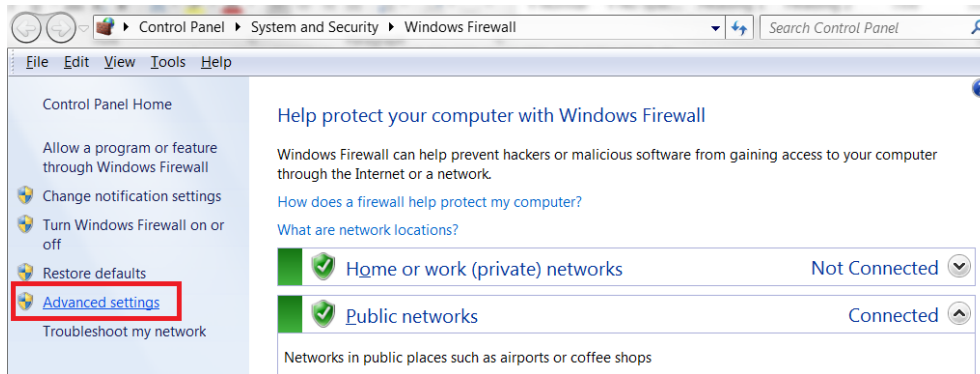
## Enabling SSH Through the Windows Firewall

Depending on network configurations, this step may not be necessary on all machines. For some machines however, the Cygwin program **sshd** will be blocked by Windows Firewall if these steps are not followed.

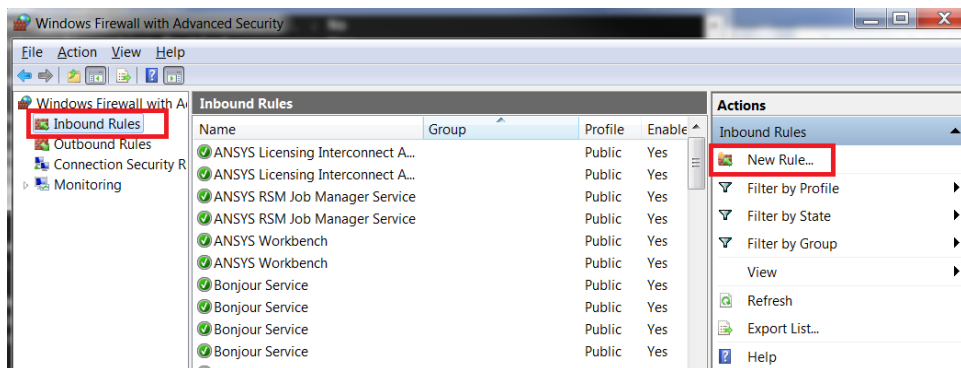
1. To allow inbound and outbound connections with Local Machine A, Remote Machine B's Windows Firewall may need to be configured to allow communication. On Remote Machine B, open the Windows Firewall from the Control Panel.



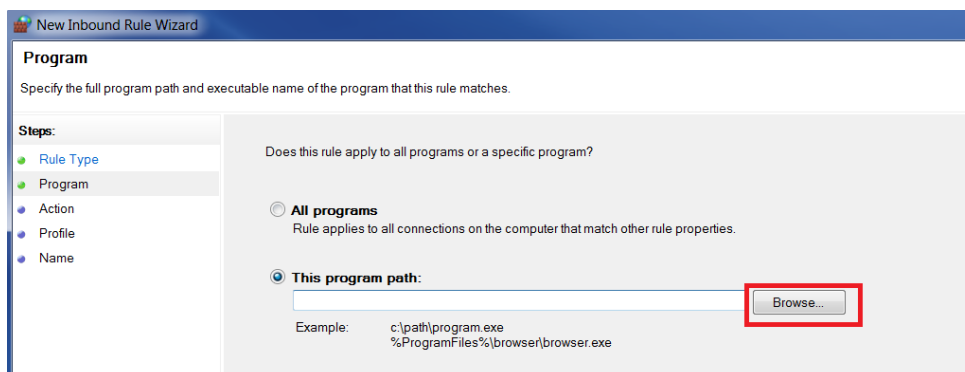
2. After Windows Firewall opens, select **Advanced settings**:



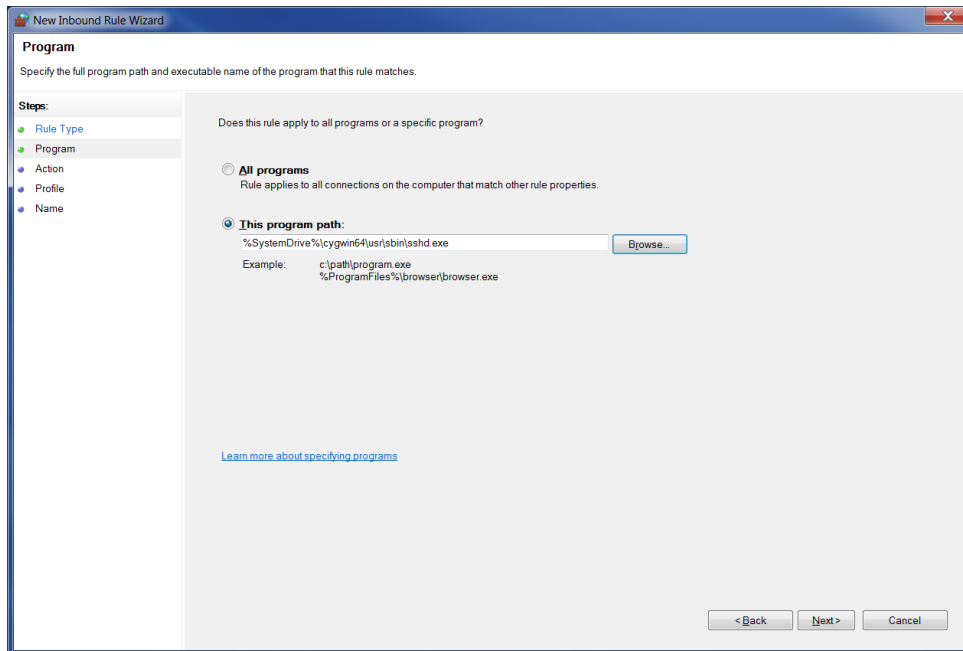
3. Select **Inbound Rules** and define a **New Rule**:



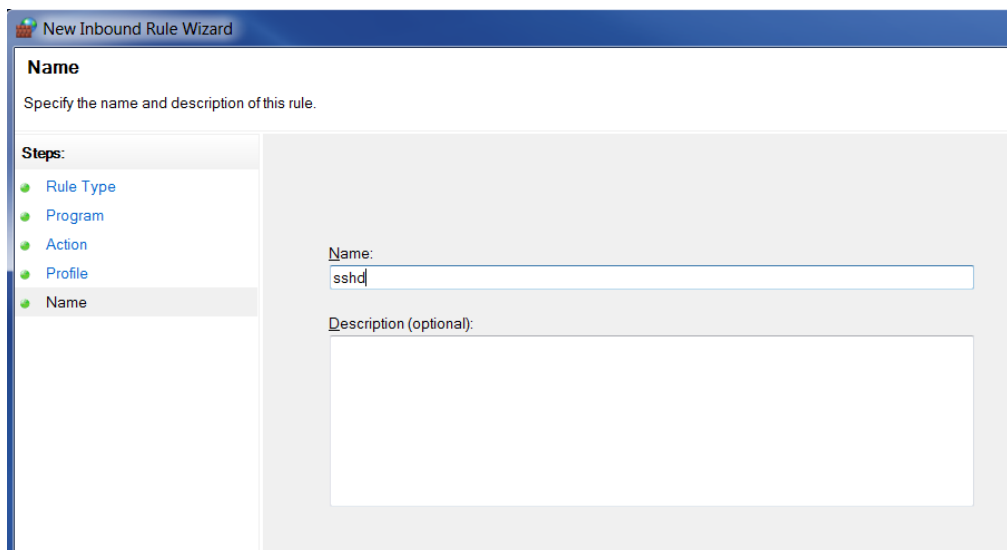
4. In the **New Inbound Rule Wizard**, select **Next** to create a rule of the default type (for example, Program). When prompted to specify the full program path and executable name of the program for this rule, click the **Browse** button:



5. Browse to the Cygwin installation directory (default location is **C:\cygwin64**) and navigate into the **usr\sbin** directory. Select the executable **ssh.exe** and click **Open**:



6. Click **Next** through the rest of the **New Inbound Rule Wizard** until you get to the **Name** of the rule. Create the name **ssh** for the rule and click **Finish**.



7. Repeat this process for an Outbound Rule.

# On Local Machine A

Local Machine A



## Installing PuTTY

PuTTY is the file transfer application recommended by the HEEDS team to serve as the SSH client on Local Machine A.

1. To download PuTTY: navigate to <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html> and choose the Windows installer as shown next.

### Binaries

#### The latest release version (beta 0.65)

This will generally be a version I think is reasonably likely to work well. If you have a problem with the release version, it might be worth trying out the latest development snapshot (below) to see if I've already fixed the bug, before reporting it to me.

The last release was signed using the old release keys (before the 2015 rollover), so it has separate RSA and DSA signatures.

#### For Windows on Intel x86

PuTTY:	<a href="#">putty.exe</a>	(or by FTP)	(RSA sig)	(DSA sig)
PuTTYtel:	<a href="#">puttytel.exe</a>	(or by FTP)	(RSA sig)	(DSA sig)
PSCP:	<a href="#">pscp.exe</a>	(or by FTP)	(RSA sig)	(DSA sig)
PSFTP:	<a href="#">psftp.exe</a>	(or by FTP)	(RSA sig)	(DSA sig)
Plink:	<a href="#">plink.exe</a>	(or by FTP)	(RSA sig)	(DSA sig)
Pageant:	<a href="#">pageant.exe</a>	(or by FTP)	(RSA sig)	(DSA sig)
PuTTYgen:	<a href="#">puttygen.exe</a>	(or by FTP)	(RSA sig)	(DSA sig)

#### A .ZIP file containing all the binaries (except PuTTYtel), and also the help files

Zip file:	<a href="#">putty.zip</a>	(or by FTP)	(RSA sig)	(DSA sig)
-----------	---------------------------	-------------	-----------	-----------

#### A Windows installer for everything except PuTTYtel

Installer:	<a href="#">putty-0.65-installer.exe</a>	(or by FTP)	(RSA sig)	(DSA sig)
------------	--	-------------	-----------	-----------

#### Checksums for all the above files

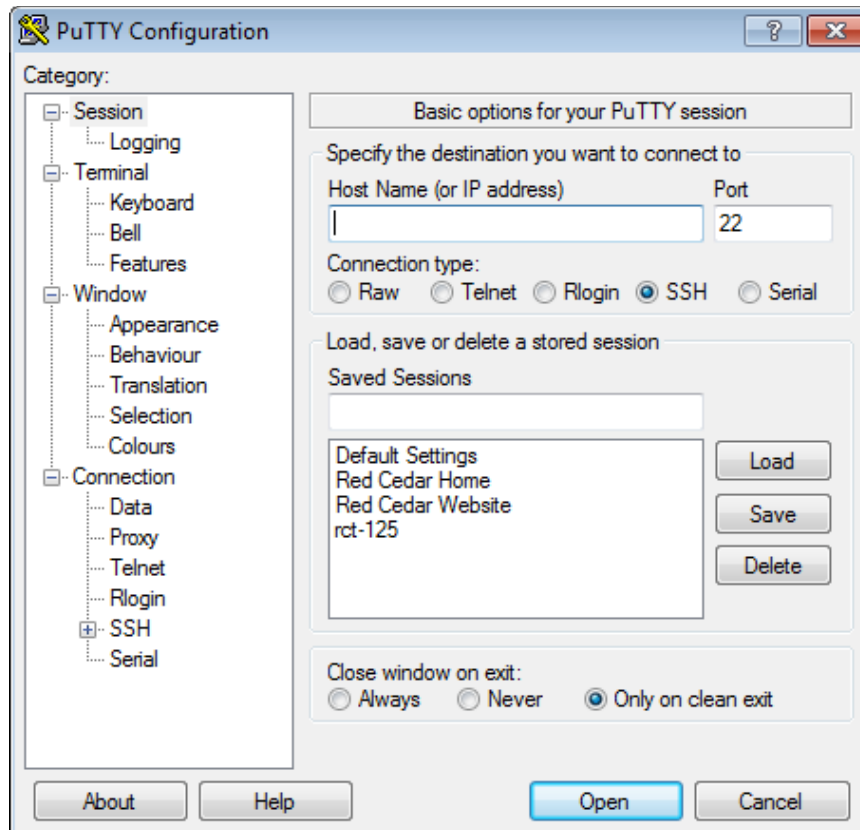
MD5:	<a href="#">md5sums</a>	(or by FTP)	(RSA sig)	(DSA sig)
SHA-1:	<a href="#">sha1sums</a>	(or by FTP)	(RSA sig)	(DSA sig)
SHA-256:	<a href="#">sha256sums</a>	(or by FTP)	(RSA sig)	(DSA sig)
SHA-512:	<a href="#">sha512sums</a>	(or by FTP)	(RSA sig)	(DSA sig)

2. Run the downloaded installer with all of the default settings.

# Testing the Connection Between Machines

Having installed PuTTY on Local Machine A and started the SSH Server on Remote Machine B, we should now be able to communicate between the two machines.

1. On local Machine A, launch PuTTY by navigating to **Start > All Programs > PuTTY > PuTTY**.
2. If you have successfully started a privileged server on Remote Machine B, you should now be able to establish an SSH connection to that machine via PuTTY. In the **Host Name (or IP address)** field, enter either the host name or the IP address of Remote Machine B, and click **Open**. If the host name does not work, use the IP address.



3. Log in using the user name and password needed to log in to Remote Machine B.

Now that you have created a communication link between the two Windows workstations, it is necessary to establish a method of password-less authentication entry so that a login prompt doesn't appear every time that you want to access the remote Windows workstation. The steps required to support this password-less entry are described in the next section.

# Creating Password-less Entry to Remote Machine B using RSA Keys

PuTTY also serves the purpose of securely passing data to Remote Machine B via RSA keys (password-less login). In order for HEEDS to exchange data with Remote Machine B, it is required that HEEDS not get prompted for a password, which PuTTY facilitates.

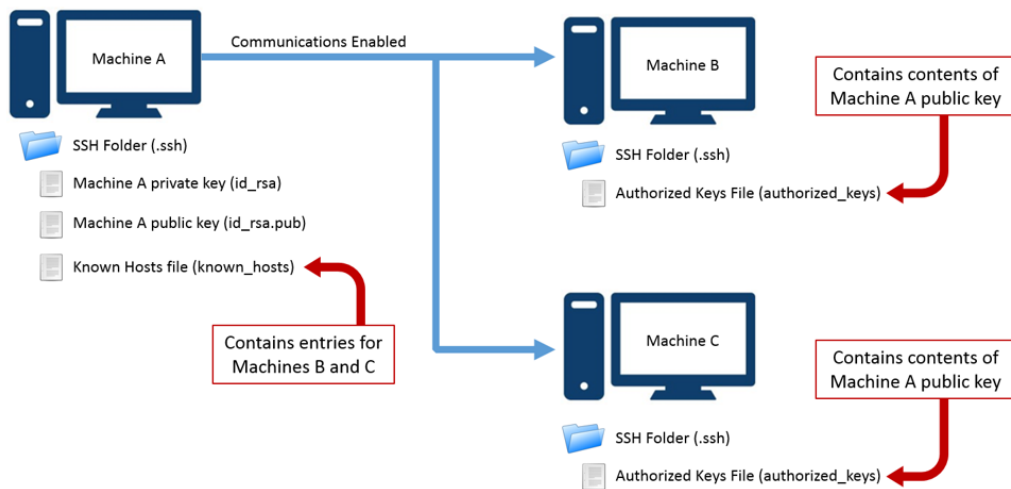
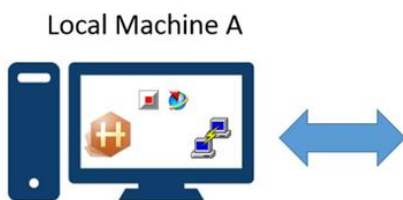


Figure 1: Public and Private Key File Setup

Figure 1 shows the standard names and locations of folders and files generated during this process. In the example shown, a user would be able to login with a password from Machine A to Machine B and from Machine A to Machine C. The important points to know are:

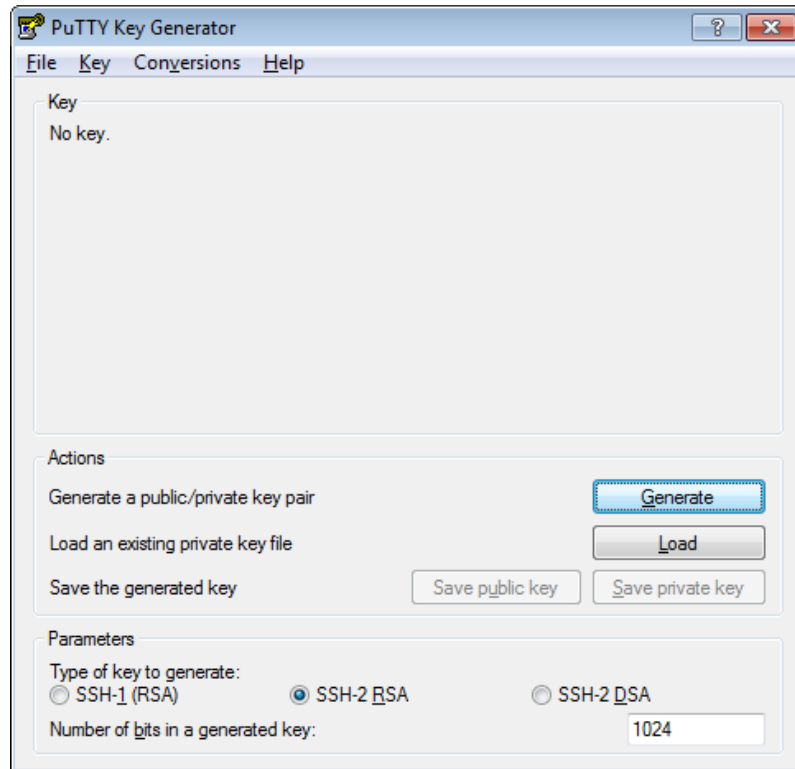
- The private key for Machine A never leaves Machine A
- The public key for Machine A will need to be included in the `authorized_key` file for any machine it will be connecting to
- The first time a connection is made, the user is prompted to accept the authenticity of the host machine. If accepted, this creates an entry in the `known_hosts` file on Machine A. This must be done manually for each host machine before trying to connect using HEEDS.

## Create Private and Public Keys on Local Machine A



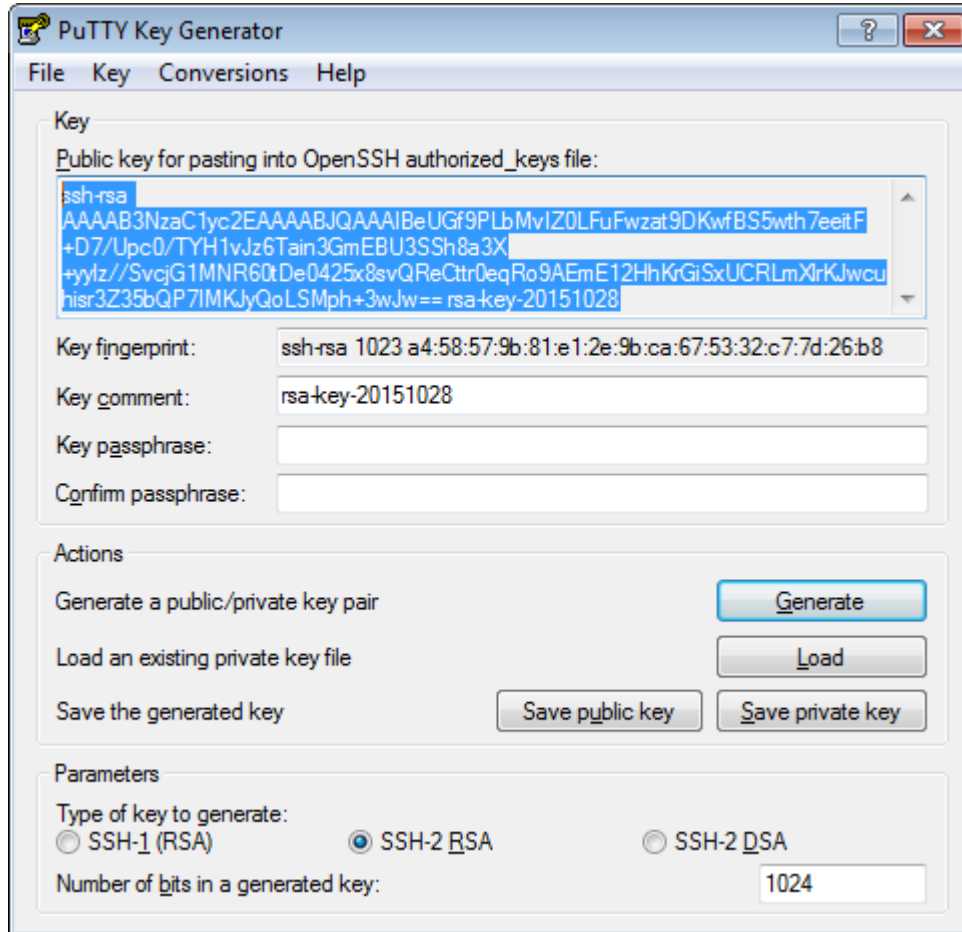
1. On local Machine A, go to **Start > All Programs > PuTTY > PuTTYgen**.

2. In PuTTYgen, click **Generate** and follow the instructions by moving your mouse around the blank area to produce an encrypted key.



3. Once the key is created, select **Save private key** and save it to a known location that you will access later. Optionally, provide a password to save with the key. The private key will be the enabler on the local machine to get access to the remote machine.

4. Copy all of the contents in the Public key field by first clicking and dragging to highlight the contents of the window and then press Ctrl+C (or right-click and **Copy**). We will place it on Remote Machine B such that when Local Machine A tries to gain access to Remote Machine B, it is granted without being prompted for a password. Note that the **Save public key** button should not be used as its format is a bit off.

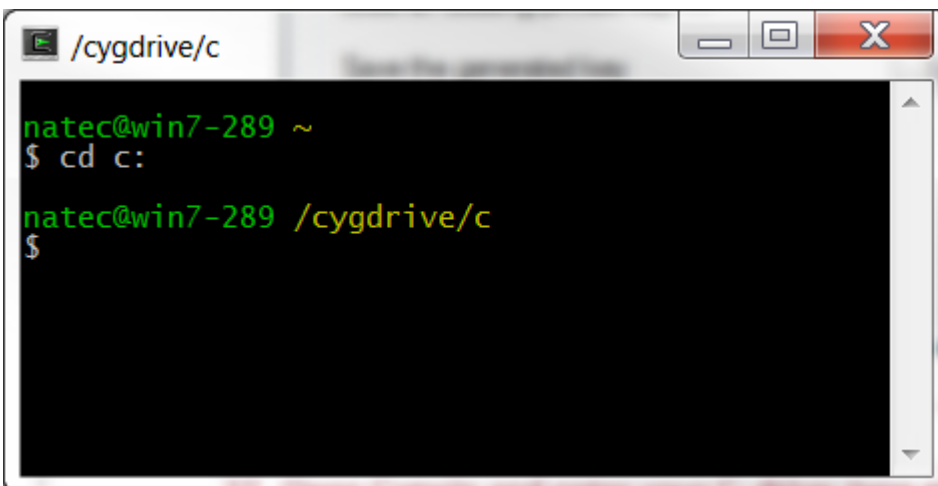


## Copy Public Key to Remote Machine B

---

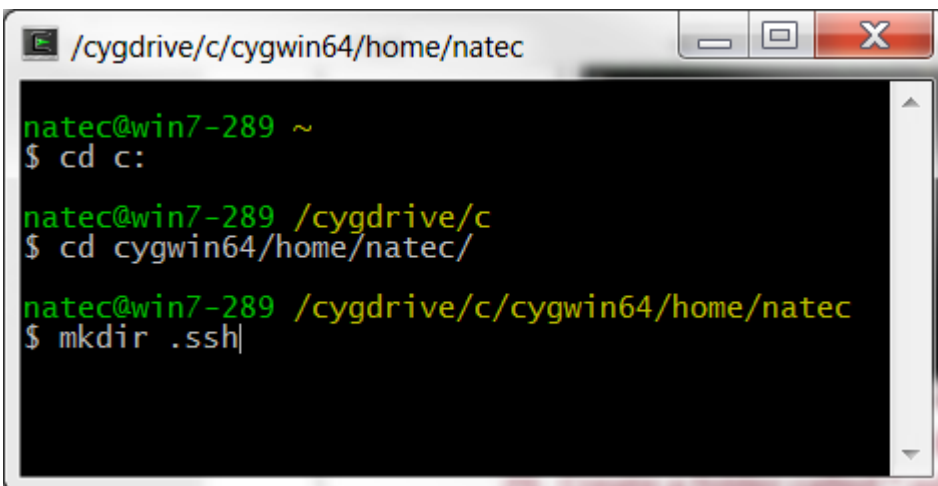


1. On remote Machine B, open Cygwin and access the drive that Cygwin was installed on (in this case we assume the C: drive) by typing **cd c:** and pressing **Enter**.



```
natec@win7-289 ~  
$ cd c:  
natec@win7-289 /cygdrive/c  
$
```

2. Change the directory to your user home directory by entering the command **cd cygwin64/home/username** and pressing **Enter**. Note that you need to provide your own username here.
3. Create a folder called **.ssh** by typing the command **mkdir .ssh** and pressing **Enter**.



```
natec@win7-289 ~  
$ cd c:  
natec@win7-289 /cygdrive/c  
$ cd cygwin64/home/natec/  
natec@win7-289 /cygdrive/c/cygwin64/home/natec  
$ mkdir .ssh|
```

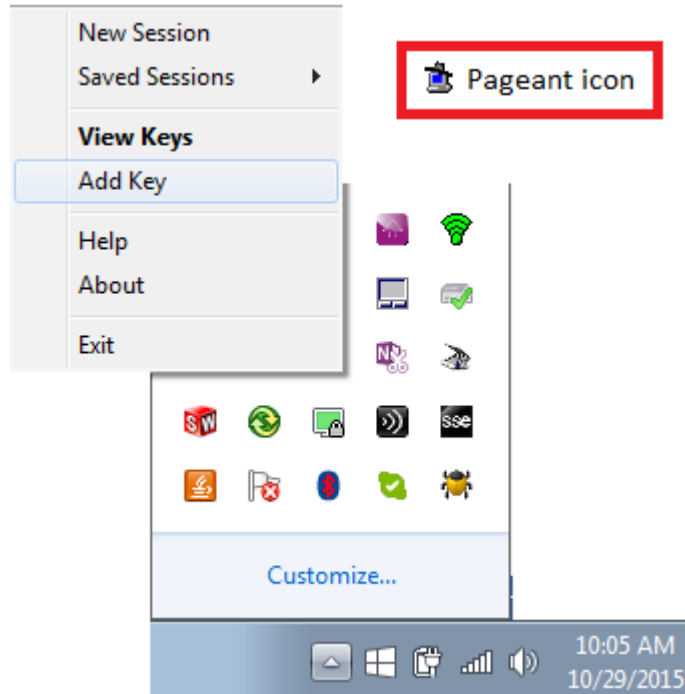
4. Create a new file named **authorized\_keys** in the **.ssh** directory. Copy your Public key (refer to step 4 of the previous section) into this file. Ensure that the file's contents begin with **ssh-rsa** (which can get lost in copying), and that it consists of only a single line of text when viewed without wrapping. If done correctly, the next section will enable the two machines to communicate without passwords.

## Enable and Test RSA Key Access with Pageant on Local Machine A

Local Machine A



1. On Local Machine A, open Pageant (**Start > All Programs > PuTTY > Pageant**). Pageant is a PuTTY SSH authentication agent. Pageant becomes active in the Windows Notification Area in the bottom right-hand corner of your screen. Right-click and select **Add Key**.



2. Navigate to the private key that you saved in PuTTYgen (step [4](#) in *Create Private and Public Keys on Local Machine A*) and click **Open**.
3. On Local Machine A, close and reopen PuTTY and log in to Remote Machine B as before (use the steps in [Testing the Connection Between Machines](#)). You will be prompted to enter a username but you should be able to log in without the need for a password. If successful, this indicates that the SSH Server-Client relationship is established and the RSA keys are working as intended. If password-less entry does not work HEEDS cannot communicate with Remote Machine B; review the above steps. If still unsuccessful consult the Troubleshooting section.



# Setting up Remote Execution in HEEDS

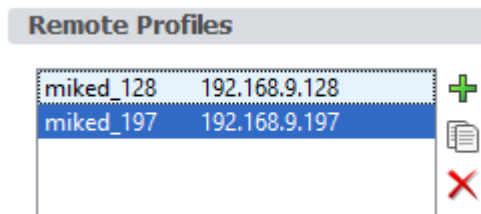
To this point, you have created a communication protocol between the two Windows workstations and allowed proper passwordless authentication. Now, you need to instruct HEEDS on how to make use of this information via a compute resource to perform design studies.



## Creating a Remote Profile

A remote profile provides HEEDS with the information it needs to connect to a remote machine. In defining a Remote Profile you will furnish HEEDS with the necessary information for data transfer to Remote Machine B.

1. Open HEEDS on Local Machine A and navigate to **File > Options > Remote Profiles**. Click the **+** button to add a new Remote Profile. Change the name of the profile to something relevant (the machine name or IP address of Remote Machine B is a good practice).



2. Next, fill in the relevant information for the newly created Remote Profile as described below:
  - a. In the remote host name and remote user name fields, you should fill in the information used to login to Remote Machine B. Ensure to pick Linux for the remote machine's operating system (Cygwin is a Linux emulator), and Cygwin as its server.
  - b. Next, define the SSH command. For PuTTY, this is the plink.exe file that comes with the installation.
  - c. Now, fill in the SCP command to point to the pscp.exe program found in the install location of PuTTY.
  - d. Lastly, choose **Specify remote location to store study data**. This is where HEEDS copies data for use in execution of analyses on the remote machine. This is a Windows path preceded by the Cygwin root, formatted for Linux. For example, the path **C:\Temp\HEEDSProjects** translates into **/cygdrive/c/Temp/HEEDSProjects**.

This completes the definition of the Remote Profile. By making use of PuTTY and Pageant on Local Machine A and Cygwin on Remote Machine B, HEEDS can access Remote Machine B, copy relevant files, execute analysis tools, and copy results back to Local Machine A

To confirm that the Remote Profile is working, use the **Test** button shown at the bottom of the configuration panel. This will attempt to connect to the remote machine, copy a small file, and execute a command to ensure that the setup works during a HEEDS study.

## Creating and Assigning a Compute Resource

Compute Resources use Remote Profile(s) to connect to the remote machines they define. They can be coupled with Schedulers or multiple Remote Profiles to best make use of the resources available to you.

1. Create a compute resource for Remote Machine B. In HEEDS, navigate to **File > Options > Compute Resources**. Click **+** to add a Direct Cluster compute resource. Name the compute resource, check the **Remote Execution** box, and in the **Profile** menu, select the newly created remote profile defined in step 1 in *Creating a Remote Profile in HEEDS MDO*.

2. On the **Process** tab of each analysis to be run on Remote Machine B choose its **Execution** tab, and select the newly created **Compute Resource** in the drop-down. The execution commands need to be modified as Cygwin is a Linux emulator. For example, when running an executable in a relative path, the prefix **./** may have to be added (e.g., **Cbeam** changes to **./Cbeam**). When a full path is given, this isn't necessary.

## Notes

---

While this process only needs to be set up one time, you will need to re-host the private key in Pageant if the Local Machine A is restarted. It is good practice to ensure the key is being hosted before beginning a study. If the key is not hosted and Pageant is not running, HEEDS cannot establish a password-less connection to the remote machine.

## SSH Doesn't Connect to Remote Machine

This section is applicable if you are not able to login to your remote machine.

1. Ensure that you are using the fully qualified domain name. For example, if the hostname is machineA, trying using machineA.city.company.com (using your company's format as appropriate). The command 'ping' is useful for diagnosing this as shown below for a success ping of the HEEDS external website:

```
C:\Users\cd1rck>ping redcedartech.com

Pinging redcedartech.com [67.227.229.17] with 32 bytes of data:
Reply from 67.227.229.17: bytes=32 time=14ms TTL=58
Reply from 67.227.229.17: bytes=32 time=12ms TTL=58
Reply from 67.227.229.17: bytes=32 time=12ms TTL=58

Ping statistics for 67.227.229.17:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 12ms, Maximum = 14ms, Average = 12ms
```

2. If you are still unable to connect, try temporarily disabling the firewall on both machines. If connections are possible without the firewall, this means the inbound and outbound exceptions for ssh were setup incorrectly.

## Password-less Connection is Unsuccessful

Read this section if you have followed the instructions in the sections above and you are still being prompted for a password when trying to ssh to the remote machine. The issue is most likely in the setup of the **authorized\_keys** file.

1. If using PuTTY and Pageant, ensure that the RSA key was copied and uploaded to the authorized\_keys file successfully. Do this by ensuring the RSA key is written on a single line by itself, that it begins with ssh-rsa and that it matches the values found in PuTTY when creating the key.
2. Once this is verified, the next step is to ensure that you are the owner of this file.
  - First log in to the Remote Machine B using PuTTY and your password
  - Navigate to the .ssh folder with the command `cd ~/.ssh`
  - Enter the command "`ls -l`" to view the permissions and the ownership information (the l in the command is a lowercase L)
3. Finally, ensure that the file permissions are set to read and write for the owner only. On the same window you used for Step 2, enter the command "`ls -l`" (the l here is a lowercase L) to view permissions and ownership.

```
$ ls -l
total 13
----- 1 Administrators None 1015 Nov 10 10:14 authorized_keys
-rw----- 1 Tester      None 1675 Mar  6 2012 id_rsa
-rw-r--r-- 1 Tester      None  394 Mar  6 2012 id_rsa.pub
-rw-r--r-- 1 Tester      None 1355 Jun 25 2014 known_hosts
```

The above screenshot shows the files in the current folder. The filenames are found in the last column. The file, `authorized_keys`, is the first row. The first column of data corresponds to the permissions of the file and the second its ownership. The current permissions are set to none, as denoted by the “-----”. The correct permissions are “-rw-----”. To update the permissions, enter the command “`chmod 600 authorized_keys`”.

```
$ chmod 600 authorized_keys
$ ls -l
total 13
-rw----- 1 Administrators None 1015 Nov 10 10:14 authorized_keys
-rw----- 1 Tester None 1675 Mar 6 2012 id_rsa
-rw-r--r-- 1 Tester None 394 Mar 6 2012 id_rsa.pub
-rw-r--r-- 1 Tester None 1355 Jun 25 2014 known_hosts
```

The above screenshot shows permissions changed to both read and write, denoted by the “-rw-----”. Next, column 2 shows that current ownership of the file belongs to Administrators. This ownership needs to match the username defined in HEEDS to access Remote Machine B. In this instance, the username is Tester. To change ownership, enter the command “`chown %username% authorized_keys`”. Where `%username%` corresponds to the username, in this instance Tester.

```
$ chown Tester authorized_keys
$ ls -l
total 13
-rw----- 1 Tester None 1015 Nov 10 10:14 authorized_keys
-rw----- 1 Tester None 1675 Mar 6 2012 id_rsa
-rw-r--r-- 1 Tester None 394 Mar 6 2012 id_rsa.pub
-rw-r--r-- 1 Tester None 1355 Jun 25 2014 known_hosts
```

Now column 2 shows correct ownership and permissions to access the `authorized_keys` file. Ensure these steps are done on `.ssh` folder as well. The user should now be able to log back in through PuTTY without using their password.

## Setting up a Local Shared Drive Resource

Typical setup for shared drive resources have the device located on the remote machine. This configuration is easy to setup and documentation on this process is located in the Help manual. These instructions assume that the drive to be used has already been shared to the network. Once this is the case, open a command prompt with administrative privileges on the local machine and enter the following:

```
"C:\Program Files (x86)\PuTTY\plink.exe" remoteUserName@remoteMachineName net use
driveLetter: \\LocalNetworkMachineName\LocalDeviceDriveName
/USER:localWindowsUserName localWindowsUserPassword /P:Yes
```

In the above command, `remoteUserName` and `remoteMachineName` reflect Remote Machine B, `driveLetter` is the label used on Remote Machine B to access this drive. This will show up similar to `T:\Program...` `LocalNetworkMachineName` is the Local Machine A's network name and `LocalDeviceDriveName` is the name of the shared drive. The information after the `/USER:` flag is the Windows login information for Local Machine A. `/P:Yes` turns on persistence which will save this mapped drive information even after logging out.

```
C:\Users\domenich>"c:\Program Files (x86)\PuTTY\plink.exe" net use T: \\myMachine\myDrive /USER:domenich password /P:Yes
```

To verify this has worked, log in to the remote machine using PuTTY and type `cd /cygdrive/driveLetter`. Your PuTTY window changes directories if the map has been successful. Now, in HEEDS, use the shared drive resource and set the location to `/cygdrive/driveLetter/folderToRunHEEDS`.

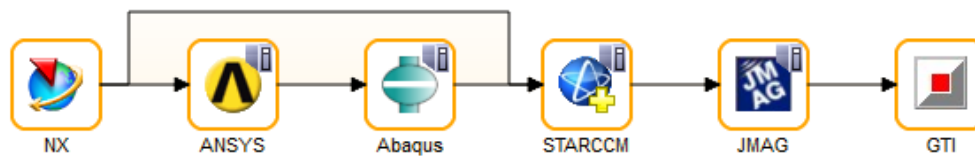
☒ Uses shared drive resource to access local study data


Shared project path:

To remove the mapped network drive, use the command:

```
driveLetter: /Delete
```

## Example Study



In the above example, both NX CAD (from Siemens PLM) and GT-Power (from Gamma Technologies, Inc.) are executed on Local Machine A, where HEEDS MDO is running. ANSYS Mechanical (from Ansys Inc.), Abaqus (from Dassault Simulia), STAR-CCM+ (from CD-adapco), and JMAG (from JSOL) are all solved remotely on Remote Machine B. This is denoted by the  image seen in the top right-hand corner of an analysis. This icon depicts the use of a direct cluster. The execution order is broken down further in the below table.

### HEEDS Example Study Execution

Execution Order	Analysis Tool	Compute Resource
1	NX CAD	Local Machine A
2	ANSYS	Remote Machine B
3	Abaqus	Remote Machine B
4	STAR-CCM+	Remote Machine B
5	JMAG	Remote Machine B
6	GT-Power	Local Machine A

## **Siemens Industry Software**

### **Headquarters**

Granite Park One 5800  
Granite Parkway  
Suite 600  
Plano, TX 75024  
USA  
+1 972 987 3000

### **Americas**

Granite Park One  
5800 Granite Parkway  
Suite 600  
Plano, TX 75024  
USA  
+1 314 264 8499

### **Europe**

Stephenson House  
Sir William Siemens Square  
Frimley, Camberley  
Surrey, GU16 8QD  
+44 (0) 1276 413200

### **Asia-Pacific**

Suites 4301-4302, 43/F  
AIA Kowloon Tower, Landmark East  
100 How Ming Street  
Kwun Tong, Kowloon  
Hong Kong  
+852 2230 3308

## **About Siemens PLM Software**

Siemens PLM Software, a business unit of the Siemens Industry Automation Division, is a leading global provider of product lifecycle management (PLM) software and services with 7 million licensed seats and 71,000 customers worldwide. Headquartered in Plano, Texas, Siemens PLM Software works collaboratively with companies to deliver open solutions that help them turn more ideas into successful products. For more information on Siemens PLM Software products and services, visit [www.siemens.com/plm](http://www.siemens.com/plm).

© 2019 Siemens Product Lifecycle Management Software Inc. Siemens and the Siemens logo are registered trademarks of Siemens AG. D-Cubed, Femap, Geolus, GO PLM, I-deas, Insight, JT, NX, Parasolid, Solid Edge, Teamcenter, Tecnomatix and Velocity Series are trademarks or registered trademarks of Siemens Product Lifecycle Management Software Inc. or its subsidiaries in the United States and in other countries. All other trademarks, registered trademarks or service marks belong to their respective holders.